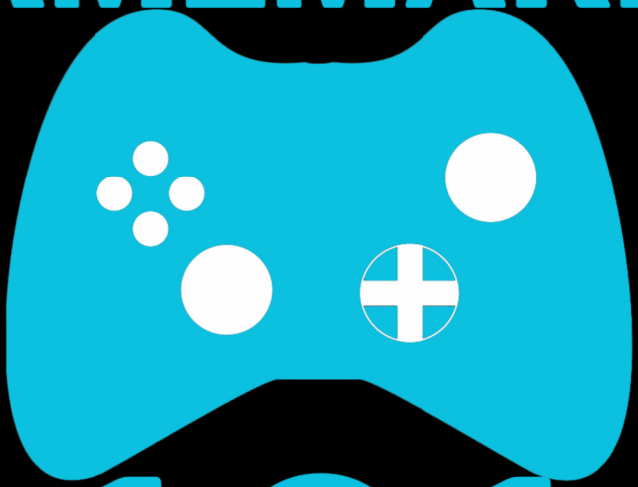
GAMEMAKER



101 MORE TIPS & TRICKS

BEN TYERS

ISBN 9798329171327

For The Download Of Project Files & Resources Please Visit:
www.GameMakerBook.com

GAMEMAKER 101 MORE TIPS & TRICKS

Table of Contents

1 Follow Object

This allows an object to follow another object, whilst keeping a distance.

Step Event:

```
if instance_exists(obj_player)
{
  if distance_to_object(obj_player)>220
  {


move_towards_point(obj_player.x,obj_player.y,4);
  }
  else
  {
       speed=0;
  }

image_angle=point_direction(x,y,obj_player.x,obj_
player.y);
}
```

2 Orbit Object

This makes one object orbit another.

Create Event:

```
ang=0;

xx=0;

yy=0;

dist=400;
```

Step Event:

```
ang++;

ang=ang mod 360;
```

Draw Event:

```
x = obj_player.x + lengthdir_x(dist, ang);

y = obj_player.y + lengthdir_y(dist, ang);

draw_self();
```

3 Random Name Generator

This code makes a random name, generated from random strings.

Code:

```
part1=choose("He","Ha","Qu","Ta","Ja","Jo","Un","
Id","Sa","Mi");

part2=choose("pad","kak","wad","pip","bar","ral",
"xen","jar");

part3=choose("au","ei","oo","ee","ou");

part4=choose("d","p","k","t","n","s");

name=part1+part2+part3+part4;
```

4 Top Down Movement

Simple top down movement and collision system.

Create Event:

```
sprite_index=spr_down;
```

Step Event:

```
if (keyboard_check(ord("W")))

{

  sprite_index=spr_up;

  image_speed=1;

  if !place_meeting(x,y-2,obj_crate) y-=2; else
y+=2;

}

else if (keyboard_check(ord("S")))

{

  sprite_index=spr_down;

  image_speed=1;

  if !place_meeting(x,y-2,obj_crate) y+=2; else
y-=2;

}



else

if (keyboard_check(ord("A")))

{

    sprite_index=spr_left;

  image_speed=1;

  if !place_meeting(x-2,y,obj_crate) x-=2;  else
x+=2;

}
```

```
else
if (keyboard_check(ord("D")))
{
    sprite_index=spr_right;
  image_speed=1;
  if !place_meeting(x+2,y,obj_crate) x+=2;  else
x-=2;
}
else
{
  image_speed=0;
}
```

5 Screen Shake

A simple screenshake effect.

Create Event:

```
viewStartX = camera_get_view_x(view_camera[0]);

viewStartY = camera_get_view_y(view_camera[0]);

screenShake = false;

shakeTimer = 0;

startAngle = 0;


//Screenshake Values - CHANGE THESE FOR
GREATER/LESSER EFFECT

shiftLeft = -5; //How much the screen may move to
the right

shiftRight = 5; //How much the screen may move to
the left

angleLeft = -1; //How much the screen angle may
tilt anti_clockwise

angleRight = 1; //How much the screen angle may
tilt clockwise

endShake = 100; //How long the shake lasts for
```

Step Event:

```
if (mouse_check_button_pressed(mb_left))

{

  screenShake = true;

  shakeTimer = 0;

}


if (screenShake == true)

{
```

```
  shakeTimer ++;

        quakeX = irandom_range(shiftRight,
shiftLeft);

  quakeY = irandom_range(shiftRight, shiftLeft);

  angle = startAngle + irandom_range(angleRight,
angleLeft);



  camera_set_view_pos(view_camera[0], 0 + quakeX,
0 + quakeY);

  camera_set_view_angle(view_camera[0], angle)

        if (shakeTimer >= endShake){

        screenShake = false;

        angle = startAngle


camera_set_view_pos(view_camera[0],viewStartX,vie
wStartY);

        camera_set_view_angle(view_camera[0], 0);

  }

}
```

6 Bomb Destruction Zone

Creates multiple explosion instances in a set pattern.

Example **Step Event**:

```
x=mouse_x;

y=mouse_y;

if mouse_check_button_pressed(mb_left)

{

  instance_create_layer(x,y,"Instances",obj_exp);

  var _callback = function()

  {

      instance_create_layer(x-
128,y,"Instances",obj_exp);

  }

  var _handle = call_later(0.5,
time_source_units_seconds, _callback);

  var _callback = function()

  {


instance_create_layer(x+128,y,"Instances",obj_exp
);

  }

  var _handle = call_later(0.5,
time_source_units_seconds, _callback);

       var _callback = function()

  {

      instance_create_layer(x,y-
128,"Instances",obj_exp);

  }

  var _handle = call_later(0.5,
time_source_units_seconds, _callback);
```

```
  var _callback = function()
  {

instance_create_layer(x,y+128,"Instances",obj_exp
);
  }
  var _handle = call_later(0.5,
time_source_units_seconds, _callback);




  var _callback = function()
  {
      instance_create_layer(x-
256,y,"Instances",obj_exp);
  }
  var _handle = call_later(1,
time_source_units_seconds, _callback);
  var _callback = function()
  {

instance_create_layer(x+256,y,"Instances",obj_exp
);
  }
  var _handle = call_later(1,
time_source_units_seconds, _callback);
       var _callback = function()
  {
      instance_create_layer(x,y-
256,"Instances",obj_exp);
  }
```

```
  var _handle = call_later(1,
time_source_units_seconds, _callback);

  var _callback = function()

  {


instance_create_layer(x,y+256,"Instances",obj_exp
);

  }

  var _handle = call_later(1,
time_source_units_seconds, _callback);




}
```

7 Circular Healthbar

This draws a circular healthbar.

Draw Event:

```
xx=500;

yy=500;

width=50;

radias=200;

start_angle=90;

amount=(360/100)*health;

//bacground

for (var i = 0; i < 360; i += 1)

{

  if i<amount draw_set_colour(c_green); else
draw_set_colour(c_red);

targetposxstart=xx+lengthdir_x(radias,i+start_ang
le);

  targetposxend=xx+lengthdir_x(radias-
width,i+start_angle);

targetposystart=yy+lengthdir_y(radias,i+start_ang
le);

  targetposyend=yy+lengthdir_y(radias-
width,i+start_angle);

draw_line_width(targetposxstart,targetposystart,t
argetposxend,targetposyend,5);

}
```

Step Event for testing:

```
if mouse_check_button(mb_left)
{
  health--;
}
if mouse_check_button(mb_right)
{
  health++;
}
health=clamp(health,0,100);
```

8 Volume Based On Distance

Changes volume based on distance between 2 instances.

Script:

```
function relerp(x0,x1,v,y0,y1)
{
    return y0+(y1-y0)*(v-x0)/(x1-x0);
}
```

Create Event:

```
dist=0;
volume=1;
min_dist=10;
max_dist=500;
music=audio_play_sound(snd_music,1,true);
```

Step Event:

```
x=mouse_x;
y=mouse_y;
dist=distance_to_object(obj_player);
volume=relerp(min_dist,max_dist,dist,1,0);
volume=clamp(volume,0,1);
audio_sound_gain(music,volume,0);
```

9 Snow Weather Effect

A simple snow effect.

Just pop this into a **Step Event**:

```
effect_create_above(ef_snow,x,y,5,c_white);
```

10 Password Easter Egg

Allows the user to type a code that could used to unlock special game features.

Create Event:

```
password="cheese";

cheat=false;
```

Step Event:

```
if keyboard_check_pressed(vk_space)

{

  keyboard_string="";

}

if keyboard_string=password

{

  cheat=true;

}
```

Example Draw Event:

```
draw_set_font(font_text);

draw_set_colour(c_white);

draw_text(50,50,"Type Password (which is
cheese");

draw_text(50,100,"Press space to clear");

draw_text(50,150,keyboard_string);


if cheat

{

  draw_text(50,250,"Cheat Unlocked");

}
```

```
else
{
  draw_text(50,250,"Cheat Locked");
}
```

11 Average Position Between 2 Instances

Gets the average position of two instances, based on their sprite origin.

Draw Event:

```
draw_set_colour(c_white);

draw_text(100,100,"Move Mouse");

xpos=obj_cross.x+(obj_target.x-obj_cross.x)/2;

ypos=obj_cross.y+(obj_target.y-obj_cross.y)/2;

draw_set_colour(c_green);

draw_circle(xpos,ypos,8,false);

draw_text(100,200,xpos);

draw_text(100,250,ypos);
```

12 Random Dice Rolls

Rolls two dice and stores the result.

Create Event:

```
dice1=0;

dice2=0;

state="wait";

value=0;
```

Step Event:

```
if state="wait" &&
mouse_check_button_pressed(mb_left)

{

  state="roll";

  alarm[0]=game_get_speed(gamespeed_fps)*4;

}

if state="roll"

{

  dice1=irandom(5);

  dice2=irandom(5);

}

if state="done" &&
mouse_check_button_pressed(mb_right)

{

  room_restart();

}
```

Alarm 0 Event:

```
state="done";
```

Draw Event:

```
draw_set_colour(c_white);

draw_text(100,100,"Doing "+state)

if state="wait" draw_text(100,150,"Tap Left
Button To Roll");


if state="roll" or state="done"

{

  draw_sprite(spr_dice,dice1,256,516);

  draw_sprite(spr_dice,dice2,640,512);

}


if state="done"

{

  value=dice1+dice2+2;//ADD 2 BECAUSE IMAGE INDEX
0 HAS A VALUE OF 1

  draw_text(100,150,"You Rolled "+string(value));

  draw_text(100,200,"Right Mouse Button To
Restart");

}
```

13 On Screen Keyboard

A simple on-screen keyboard that allows the user to enter text.

obj_conrol Create Event:

```
//top row

var alphabet = "QWERTYUIOP";

for(i = 0; i<10; i++){

    var bt = instance_create_layer(64+i*32+(8*i),
64,"Instances", obj_button);

    bt.letter = string_char_at(alphabet, i+1);

}

//middle row

var alphabet = "ASDFGHJKL";

for(i = 0; i<9; i++){

    var bt = instance_create_layer(72+i*32+(8*i),
104, "Instances",obj_button);

    bt.letter = string_char_at(alphabet, i+1);

}

//BOTTOM row

var alphabet = "ZXCVBNM";

for(i = 0; i<7; i++){

    var bt = instance_create_layer(88+i*32+(8*i),
144,"Instances", obj_button);

    bt.letter = string_char_at(alphabet, i+1);

}


global.entered_string="";
```

Step Event:

```
if keyboard_check_pressed(vk_space)
{
  global.entered_string="";
}
if keyboard_check_pressed(vk_enter)
{
  room_goto(room_show);
}
```

Draw Event:

```
draw_set_colour(c_black);
draw_set_halign(fa_left);
draw_set_font(fnt_word);
draw_text(100,400,global.entered_string);
draw_text(10,600,"Tap Space To Clear Text");
draw_text(10,650,"Tap Enter To Save");
draw_text(10,700,"Click Buttons With Mouse");
```

obj_button Create Event:

```
letter = "A";
image_speed = 0;
```

Step Event:

```
if instance_position(mouse_x,mouse_y,id)
{
    if mouse_check_button(mb_left)
  {
        image_index = 3;
```

```
    }
  else
  {
        image_index = 2;
    }


    if mouse_check_button_released(mb_left)
  {
      global.entered_string+=letter
  }
}
else
{
  image_index=0;
}
```

Draw Event:

```
draw_self();
draw_set_font(fnt_button);
draw_set_halign(fa_center);
draw_set_valign(fa_top);
draw_text(x+16, y+4,
string_hash_to_newline(letter));
draw_set_halign(fa_left);
```

14 Array Sort

This example sorts words alphabetically.

Create Event:

```
array=["Pear","Apple","Banana","Lemon","Kiwi","Me
lon","Grape","Orange"];
```

Step Event:

```
if mouse_check_button_pressed(mb_left)

{

  array_sort(array,true);

}

if mouse_check_button_pressed(mb_right)

{

  array_sort(array,false);

}

if mouse_check_button_pressed(mb_middle)

{

  room_restart();

}
```

Draw Event:

```
var size= array_length_1d(array);

draw_set_colour(c_white);

draw_set_font(font_text);

draw_text(50,50,"Current Order");

for (var i = 0; i < size-1; i += 1)

{

  draw_text(100,200+(40*i),array[i]);

}

draw_text(50,600,"Left Mouse Button To Sort Down\
nRight Mouse Button Sort Up\nMiddle Button To
Restart");
```

15 Colour Picker

A system that allows the player to click and store a colour. Great for letting the user choose a colour scheme.

Create Event:

```
saved_colour=c_red;

current_colour=c_white;
```

Step Event:

```
if mouse_check_button_pressed(mb_left)

{

  saved_colour=draw_getpixel(mouse_x,mouse_y);

}
```

Draw Event:

```
draw_sprite(sprite_index,0,room_width/2,0);

current_colour=draw_getpixel(mouse_x,mouse_y);

draw_set_font(font_text);

draw_set_colour(current_colour);

draw_text(200,700,"Current Color");

draw_set_colour(saved_colour);

draw_text(200,650,"Saved Colour - Left Mouse
Button To Save");
```

16 Numbers To Speech

This script turns numbers to speech. Range is 0 to 999999.

Script code:

```
function integer_to_english(int) {

    // Lookup tables

    static _digits_ = ["", One, Two, Three, Four,
Five, Six, Seven, Eight, Nine];

    static _teens_ = [Ten, Eleven, Twelve,
Thirteen, Forteen,Fifteen, Sixteen,
Seventy,Eighteen, Nineteen];

    static _tens_ = ["", "", Twenty, Thirty,
Forty, Fifty, Sixty, Seventy, Eighty, Ninety];

  strings=ds_list_create();

// Decompose digits



    // Decompose digits

    var thousands = (int div 1000) mod 1000;

    var hundreds = (int div 100) mod 10;

    var tens = (int div 10) mod 10;

    var units = int mod 10;


    // Accumulator

    var str = "";


    // Add thousands

    if (thousands > 0) {

       integer_to_english(thousands);

    ds_list_add(strings, Thousand);
```

```
    }

    // Add hundreds
    if (hundreds > 0) {
        if ds_list_size(strings)!=0 {

        }
         ds_list_add(strings, _digits_[hundreds])
         ds_list_add(strings,Hundred);
    }

    // Add tens and digits
    if (int mod 100 > 0) {
        if ds_list_size(strings)!=0 {
             ds_list_add(strings,And);
        }
        switch (tens) {
            case 0:

ds_list_add(strings,_digits_[units]);
            break;
            case 1:
                 ds_list_add(strings,
_teens_[units]);
            break;
            default:

ds_list_add(strings,_tens_[tens]);
                if (units > 0) {
```

```
ds_list_add(strings,_digits_[units]);
                }
            break;
        }
    }

    // Done
    return strings;
}
```

Example Usage, **obj_demo:**

Create Event:

```
value=irandom(100000);
list=ds_list_create();
list=integer_to_english(value);
voice=audio_play_sound(Pause,1,false);
```

Step Event:

```
if !audio_is_playing(voice)
{
  if ds_list_size(list)!=0
  {
       to_play=list[|0];
       voice=audio_play_sound(to_play,1,0);
       ds_list_delete(list,0);
  }

}
```

```
if mouse_check_button_pressed(mb_left)
{
  room_restart();
}
```

Draw Event:

```
draw_set_font(c_white);
draw_set_font(font_text);
draw_text(100,100,value);
size=ds_list_size(list);
draw_text(100,200,size);
```

17 Teleport

Teleports player between two instances.

Create Event:

```
sprite_index=spr_down;
```

Step Event:

```
if (keyboard_check(ord("W")))
{
  sprite_index=spr_up;
  image_speed=1;
  y-=2;
}
else if (keyboard_check(ord("S")))
{
  sprite_index=spr_down;
  image_speed=1;
  y+=2;
}

else
if (keyboard_check(ord("A")))
{
    sprite_index=spr_left;
  image_speed=1;
  x-=2;
}
else
if (keyboard_check(ord("D")))
```

```
{
    sprite_index=spr_right;
  image_speed=1;
  x+=2;
}
else
{
  image_speed=0;
}
if instance_place(x,y,obj_red)
{
  x=obj_green.x+100;
  y=obj_green.y;
}
if instance_place(x,y,obj_green)
{
  x=obj_red.x+100;
  y=obj_red.y;
}
```

18 Queued Messages

This queues text message and plays them in order.

Great to provide info to the player, for example on a tutorial level.

Create Event:

```
value="";

list=ds_list_create();

ds_list_add(list,"Hello","Sample Message","Nice
To Meet You","Have A Great Day");

alarm[0]=game_get_speed(gamespeed_fps)*4;
```

Alarm 0 Event:

```
alarm[0]=game_get_speed(gamespeed_fps)*4;

if ds_list_size(list)!=0

{

  value=list[|0];

  ds_list_delete(list,0);

}
```

DrawEvent:

```
draw_set_font(c_white);

draw_set_font(font_text);

draw_text(100,100,value);
```

19 Shop System

A simple shop system that allows player to buy and sell weapons.

obj_button_parent Step Event:

```
if position_meeting(mouse_x,mouse_y,id)
{
  image_index=1;
  if mouse_check_button(mb_left)
  {
       image_index=2;
  }
}
else
{
  image_index=0;
}
```

obj_buy_parent Left Mouse Button Release Event:

```
if global.cash>=global.items[my_id,5]
{
  global.items[my_id,3]+=1;
  global.cash-=global.items[my_id,5];
  audio_play_sound(snd_purchase,1,false);
}
else
{
  audio_play_sound(snd_not_enough,1,false);
}
```

```
obj_sell_parent Left Mouse Button Released Event:

if global.items[my_id,3]>=1//check player has
item to sell

{

  global.items[my_id,3]-=1;//reduce item counnt

  global.cash+=global.items[my_id,4];//increase
cash

  audio_play_sound(snd_cash,1,false);

}

else//no items to sell

{

  audio_play_sound(snd_no_items,1,false);

}
```

obj_buy_1 Create Event:

```
my_id=1;
```

obj_sell_1 Create Event:

```
my_id=1;
```

obj_hud Create Event:

```
global.cash=1000;


//gun1

global.items[1,1]=spr_gun_1;    //gun sprite

global.items[1,2]="Gun";        //description

global.items[1,3]=10;           //ammount

global.items[1,4]=50            //cost

global.items[1,5]=40;           //sell for


//gun2
```

```
global.items[2,1]=spr_gun_2;    //gun sprite

global.items[2,2]="Double Shotgun";
  //description

global.items[2,3]=5;            //ammount

global.items[2,4]=100           //cost

global.items[2,5]=80;           //sell for


//gun3

global.items[3,1]=spr_gun_3;    //gun sprite

global.items[3,2]="Missile Launcher";
  //description

global.items[3,3]=1;            //ammount

global.items[3,4]=1000          //cost

global.items[3,5]=800;          //sell for
```

Draw Event:

```
draw_set_font(font_info);

draw_set_colour(c_white);

draw_set_halign(fa_center);

draw_set_valign(fa_middle);


///draw cash

draw_text(room_width/2,80,"Cash Available
"+string(global.cash));


//draw headers

draw_text(128,128,"Image");

draw_text(256,128,"Type");

draw_text(384,128,"Inventory");

draw_text(512,128,"Buy For");
```

```
draw_text(640,128,"Sell For");

//draw images
draw_sprite(global.items[1,1],0,128,256);
draw_sprite(global.items[2,1],0,128,384);
draw_sprite(global.items[3,1],0,128,512);
for (var i = 1; i <= 3; i += 1)
{
  draw_sprite(global.items[i,1],0,128,128+128*i);
}

//draw info
for (var i = 1; i <= 3; i += 1)
{
  for (var j = 2; j <= 5; j += 1)
  {

  draw_text(128*j,128+(128*i),global.items[i,j]);
  }
}
```

20 Parallax Background

A simple example of how multiple layers can be used to create a parallax background.

Create Event:

```
flying_level=200;
```

Step Event:

```
/// @description Object management

global.difference=(flying_level-y)/10;//set as a
global value as it will be used for parallax
background

image_angle=global.difference-10;

//keep in screen

y=clamp(y,20,380)

//move

if y<flying_level y+=2;

if y>flying_level y-=2;

//backgrounds

var lay_id1 = layer_get_id("bbg_1");

var lay_id2 = layer_get_id("bbg_2");

var lay_id3 = layer_get_id("bbg_3");

layer_y(lay_id1,-200+global.difference*2);

layer_y(lay_id2,30+global.difference*1);

layer_y(lay_id3,40+global.difference*0.5);


if keyboard_check(ord("W")) y-=5;

if keyboard_check(ord("S")) y+=5;
```

Draw GUI:

```
draw_set_colour(c_black);

draw_set_font(font_text);

draw_text(20,20,"Use Keys W & S");
```

21 Eight Directional Movement

8 Directional sprite control, with basic movement.

Step Event:

```
up=keyboard_check(ord("W"))
down=keyboard_check(ord("S"))
left=keyboard_check(ord("A"))
right=keyboard_check(ord("D"))

if up && right
{
  sprite_index=spr_up_right;
  image_speed=1;
  x=x+lengthdir_x(1,45);
  y=y+lengthdir_y(1,45);
}
else
if up && left
{
  sprite_index=spr_up_left;
  image_speed=1;
  x=x+lengthdir_x(1,135);
  y=y+lengthdir_y(1,135);
}
else
if down && left
{
  sprite_index=spr_down_left;
```

```
  image_speed=1;
  x=x+lengthdir_x(1,225);
  y=y+lengthdir_y(1,225);
}
else
if down && right
{
  sprite_index=spr_down_right;
  image_speed=1;
  x=x+lengthdir_x(1,315);
  y=y+lengthdir_y(1,315);
}
else
if up
{
  sprite_index=spr_up;
  image_speed=1;
  y--
}
else
if down
{
  sprite_index=spr_down
  image_speed=1;
  y++;
}
else
if left
```

```
{
  sprite_index=spr_left
  image_speed=1;
  x--
}
else
if right
{
  sprite_index=spr_right;
  image_speed=1;
  x++;
}
else
{
  image_speed=0;
}
```

22 Room Transition Fade In & Out

Darkens the room on room start and changing rooms.

obj_fade_in Create Event:

```
alp=0;
active=false;
```

Step Event:

```
if active
{
  alp+=0.01;
  if alp>1 room_goto(Room1);
}
```

Draw Event:

```
draw_set_colour(c_black);
if active
{
  draw_set_alpha(alp);

draw_rectangle(0,0,room_width,room_height,false);
  draw_set_alpha(1);
}
obj_fade_out Create Event:
alp=0;
active=false;
```

Step Event:

```
if active
{
  alp+=0.01;
```

```
  if alp>1 room_goto(Room2);
}
if mouse_check_button_pressed(mb_left)
{
  active=true;
}
```

Draw Event:

```
draw_set_colour(c_black);
if active
{
  draw_set_alpha(alp);

draw_rectangle(0,0,room_width,room_height,false);
  draw_set_alpha(1);
}
draw_set_font(font_text);
draw_text(50,50,"Tap Left Mouse Button To Change
Rooms");
```

23 Nine Slice Example

Draws an image with special settings, allowing various size boxes for example.

Assumes nine slice sprite has been set up.

Draw Event:

```
draw_sprite_ext(spr_example, 0,
100,100,3,1,0,c_white,1);


draw_sprite_ext(spr_example, 0,
100,400,2,3,0,c_white,1);
```

24 Hit Box

Makes a hit box when attacking that can be used to detect attacks.

obj_player Create Event:

```
image_speed=0;
```

Step Event:

```
if mouse_check_button_pressed(mb_left)

{

  image_speed=1;

}

if image_index=4

{

  instance_create_layer(x+220,y+70,"hit",
obj_hit);

}

if image_index=image_number-1

{

  image_speed=0;

  image_index=0;

}
```

obj_hit Create Event:

```
alarm[0]=game_get_speed(gamespeed_fps);
```

Alarm 0 Event:

```
instance_destroy();
```

25 Snap To Grid

Snaps an instance to a grid and allows placements of instances at those positions.

Step Event:

```
x=mouse_x;

y=mouse_y;

move_snap(64,64);

if mouse_check_button_pressed(mb_left)

{


instance_create_layer(x,y,"Instances",obj_placed)
;

}
```

26 Hide & Seek

A enemy that moves towards the player if they can see it.

obj_enemy Create Event:

```
can_see=false;
```

Step Event:

```
if
collision_line(x,y,obj_player.x,obj_player.y,obj_
crate, false, false)

{

  can_see=false;

}

else

{

  can_see=true;

}

if can_see

{


move_towards_point(obj_player.x,obj_player.y,1);

}

else

{

  speed=0;

}
```

27 Save Highscore

A simple system to save a highscore.

To load current highscore:

```
ini_open("savedata.ini");
highscore = ini_read_real("save1", "score", 0 );
ini_close();
```

To test and save:

```
if score>highscore
  {
       ini_open("savedata.ini");
       ini_write_real("save1", "score", score );
       ini_close();
       game_restart();
  }
```

28 Sprite Animation Control

A simple system to play an animation and then return to idle when animation is complete.

Create Event:

```
sprite_index=spr_idle;
image_speed=1;
```

Step Event:

```
if keyboard_check(ord("A"))
{
  image_index=0;
  sprite_index=spr_attack;
}
if keyboard_check(ord("D"))
{
  image_index=0;
  sprite_index=spr_dead;
}
if keyboard_check(ord("H"))
{
  image_index=0;
  sprite_index=spr_hurt;
}
if keyboard_check(ord("J"))
{
  image_index=0;
  sprite_index=spr_jump;
}
```

```
if keyboard_check(ord("R"))
{
  image_index=0;
  sprite_index=spr_run;
}
if keyboard_check(ord("W"))
{
  image_index=0;
  sprite_index=spr_walk;
}
```

29 Jet Pack & Gravity

Flying control system.

Create Event:

```
gravity=0.1;

image_angle=45;
```

Step Event:

```
if mouse_check_button(mb_left)
{
  motion_add(90,0.5);
}
if y<128
{
  vspeed=0;
  y=129;
}
if y>room_height-128
{
  vspeed=0;
  y=room_height-129;
}
```

30 Enemy Movement On Platform

Make an enemy walk on turn whilst on a platform.

Create Event:

```
move_speed=2;
```

Step Event:

```
x+=move_speed;

xpos=x+sign(move_speed)*40;

ypos=y+5;

if !instance_position(xpos,ypos,obj_platform)

{

  move_speed=-move_speed;

}

image_xscale=sign(move_speed);
```

31 Tweening

This slowly moves an instance between two points with speed based on the distance remaining.

Create Event:

```
dir=1;

start=200;

target=800;

x=start;
```

Step Event:

```
if dir==1

{

  diff=target-x;

  x+=diff/12;

}

if dir=-1

{

  diff=start-x;

  x+=diff/12;

}

if x>=target-1

{

  dir=-1;

}

if x<=start+1

{

  dir=1;

}
```

32 Door & Key

A system that only allows a player open a door when they have the key.

Create Event:

```
has_key=false;
```

Collision With obj_key:

```
has_key=true;

with other instance_destroy()
```

Collision With obj_door:

```
if has_key

{

  with other instance_destroy();

}

else

x=xprevious;

y=yprevious;
```

Step Event:

```
if (keyboard_check(ord("W")))

{

  sprite_index=spr_up;

  image_speed=1;

  y-=2;

}

else if (keyboard_check(ord("S")))

{

  sprite_index=spr_down;
```

```
  image_speed=1;
  y+=2;
}

else
if (keyboard_check(ord("A")))
{
    sprite_index=spr_left;
  image_speed=1;
  x-=2;
}
else
if (keyboard_check(ord("D")))
{
    sprite_index=spr_right;
  image_speed=1;
  x+=2;
}
else
{
  image_speed=0;
}
```

33 Wrap Instance Around Room

Wraps an object around room border, drawing around edges of the room.

Draw Event example:

```
draw_self();

if (keyboard_check(ord("A")))

{

  x-=2;

}


if (keyboard_check(ord("D")))

{

  x+=2;

}

if x<0 x=room_width;

if x>room_width x=0;

size=sprite_width/2;

if x<size*2

{

  draw_sprite(sprite_index,0,room_width+x-
size+116,y);

}

if x>room_width-size*2

{

  draw_sprite(sprite_index,0,size*2-(pos)-256,y);

}
```

34 Change Transparency On Collision

A useful idea, that can used when a player goes beneath a tree or building roof.

Tree Create Event:

```
alp=1;
```

Step Event:

```
if
instance_position(obj_player.x,obj_player.y,id)

{

  alp=0.4;

}

else

{

  alp=1;

}
```

Draw Event:

draw_sprite_ext(sprite_index,0,x,y,1,1,0,c_white,alp);

35 Weapon Upgrade System

A simple system to allow player to upgrade weapons.

Weapon Create Event:

```
number_of_weapons=image_number;

current=0;

alarm[0]=game_get_speed(gamespeed_fps)*4;
```

Step Event:

```
if keyboard_check_pressed(ord("U"))

{

  current++;

}

if keyboard_check_pressed(ord("D"))

{

  current--;

}

current=clamp(current,0,number_of_weapons-1);
```

Alarm 0 Event:

```
alarm[0]=game_get_speed(gamespeed_fps)*4;

if current=0

{

  instance_create_layer(x,y,"bullets",obj_bullet,
{hspeed:5});

}

if current=1

{


instance_create_layer(x,y+22,"bullets",obj_bullet
,{hspeed:5});
```

```
  instance_create_layer(x,y-
22,"bullets",obj_bullet,{hspeed:5});

}

if current=2

{


instance_create_layer(x,y+8,"bullets",obj_bullet,
{hspeed:5});

  instance_create_layer(x,y-
8,"bullets",obj_bullet,{hspeed:5});

}

if current=3

{


instance_create_layer(x,y+16,"bullets",obj_bullet
,{hspeed:5});

  instance_create_layer(x,y-
16,"bullets",obj_bullet,{hspeed:5});

}

if current=4

{


instance_create_layer(x,y+16,"bullets",obj_missil
e,{hspeed:5});

  instance_create_layer(x,y-
16,"bullets",obj_missile,{hspeed:5});

}

if current=5

{


instance_create_layer(x,y,"bullets",obj_missile,
{hspeed:5});
```

```
instance_create_layer(x,y+16,"bullets",obj_missil
e,{hspeed:5});

  instance_create_layer(x,y-
16,"bullets",obj_missile,{hspeed:5});



}
```

Draw Event:

```
draw_sprite(sprite_index,current,x,y);
```

36 Knockback

A basic knockback system that can built upon.

Create Event:

```
knockback=false;

target=1200;
```

Step Event:

```
if x<target &&  knockback=false
{
  x++;
  image_speed=1;
}
else
{
  image_speed=0;
}
if mouse_check_button_pressed(mb_left)
{
  knockback=true;
  image_index=5;
  alarm[0]=game_get_speed(gamespeed_fps)*2;
}
if knockback
{
  x-=2;
}
```

Alarm 0 Event:

```
knockback=false;
```

37 Road Builder

Allows player to draw a path. Great for a range of game genres.

Create Event:

```
can_build=false;

cursor_x=0;

cursor_y=0;
```

Step Event:

```
var road;

road = obj_road;

cursor_x = (mouse_x div 32)* 32

cursor_y = (mouse_y div 32)* 32

can_build =

(

    place_meeting(cursor_x - 1, cursor_y, road)
||

    place_meeting(cursor_x + 1, cursor_y, road)
||

    place_meeting(cursor_x, cursor_y + 1, road)
||

    place_meeting(cursor_x, cursor_y - 1, road) )

    && !place_meeting(cursor_x, cursor_y, road)

if mouse_check_button_pressed(mb_left) &&
can_build

{ instance_create_layer(cursor_x,cursor_y,"Instan
ces",obj_road); }
```

Draw Event:

```
draw_sprite(spr_cursor,can_build,cursor_x,cursor_
y);
```

38 Select Multiple Troops

Allows selection of multiple instances.

Control Object:

Create Event:

```
x1=mouse_x;
y1=mouse_y;
x2=mouse_x;
y2=mouse_y;
list=ds_list_create();
```

Step Event:

```
if mouse_check_button(mb_left)
{
  x2=mouse_x;
  y2=mouse_y;
}
if mouse_check_button_pressed(mb_left)
{
  x1=mouse_x;
  y1=mouse_y;
  with obj_troop
  {
       selected=false;
  }
  ds_list_clear(list);
}
if mouse_check_button_pressed(mb_right)
```

```
{
  x1=mouse_x;
  y1=mouse_y;
  x2=mouse_x;
  y2=mouse_y;
  with obj_troop
  {
       selected=false;
  }
  ds_list_clear(list);
}
if mouse_check_button_released(mb_left)
{
  ds_list_clear(list)
  with obj_troop
  {
       if !
collision_rectangle(other.x1,other.y1,other.x2,ot
her.y2,id,false,false) continue
       {
            id.selected=true;
            ds_list_add(other.list,id);
       }
  }
}
```

Draw Event:

```
draw_set_colour(c_green)
draw_rectangle(x1,y1,x2,y2,true);
```

```
draw_text(800,50,"Selected IDs");

if ds_list_size(list)>0

{

  for (i=0;i<ds_list_size(list);i++;)

  {

       draw_text(800,100+(i*32),list[|i]);

  }

}

draw_text(50,50,"Left Mouse Click & Drag To
Select - Right Click Reset");
```

obj_troop

Crete Event:

```
selected=false;
```

Draw Event:

```
if selected

{

  draw_set_colour(c_lime);


draw_rectangle(bbox_left,bbox_top,bbox_right,bbox
_bottom,true);


draw_sprite_ext(sprite_index,image_index,x,y,1,1,
0,c_red,1);

}

else

{

  draw_self();

}
```

39 Road Connections

System automatically draw the correct subimage to make roads connect.

The main code tests for surrounding road connections and sets the according subimage:

```
var t, b, l, r, d, j;

j = object_index;

d = 48;//size

t = place_meeting(x, y - d, j);

b = place_meeting(x, y + d, j);

r = place_meeting(x + d, y, j);

l = place_meeting(x - d, y, j);

image_index = l + r * 2 + t * 4 + b * 8;
```

See the example program for a build version,

40 Lightning Effect

A simple lightning effect that is adaptable for a range of uses.

Script:

```
function draw_lightning_simple(xpos1, ypos1,
xpos2, ypos2, size1, size2)

{

  var xprev,yprev,xx,yy,dir,t;

  xx=xpos1;

  yy=ypos1;

  t=median(-89,size2,89)

  do

  {

      xprev=xx;

       yprev=yy;


dir=point_direction(xprev,yprev,xpos2,ypos2)+(t-
random(t*2));

      xx+=lengthdir_x(size1,dir);

      yy+=lengthdir_y(size1,dir);

      draw_line(xprev,yprev,xx,yy);

  }

  until
(point_distance(xx,yy,xpos2,ypos2)<=size1)

  draw_line(xx,yy,xpos2,ypos2);

}
```

Example Usage in **Draw Event**:

```
draw_lightning_simple(x,y,mouse_x,mouse_y,32,60);
```

41 Gravity Movement

Showing various items with gravity.

Create Event:

```
grav=0.1;

gravity=grav;

dir="down"

has_hit=false;

vspeed-=5;
```

Step Event:

```
if position_meeting(x,y+1,obj_crate)

{

  gravity=0;

  vspeed=0;

  while instance_position(x,y+1,obj_crate)

  {

       y--;

  }

}

if mouse_check_button_pressed(mb_left) &&
instance_position(mouse_x,mouse_y,id)

{

  vspeed=-6;

  gravity=grav;

}
```

42 Blood Damage Effect

Creates a blood effect that drips down the screen.

Create Event:

```
image_index=irandom(6);

image_speed=0;

image_angle=irandom(359);

move=false;

alp=1;

alarm[0]=game_get_speed(gamespeed_fps)*2;
```

Step Event:

```
if move

{

    alp-=0.004;

    y++;

    if alp<=0 instance_destroy();

}
```

Alarm 0 Event:

```
move=true;
```

Draw Event:

```
draw_sprite_ext(sprite_index,image_index,x,y,1,1,
image_angle,c_white,alp);.
```

43 Tap Instance To Change Image

Tap an instance to change it’s subimage.

Create Event:

img=0;

Step Event:

```
if mouse_check_button_pressed(mb_left) &&
instance_position(mouse_x,mouse_y,id)

{

  img++;

  if img=image_number img=0;

}
```

Draw Event:

```
draw_sprite(sprite_index,img,x,y);
```

44 Bullet Holes

A simple system for showing bullet shots. As a bonus the bullet holes can be used to detect damage.

Control Object Step Event:

```
x=mouse_x;

y=mouse_y;

if mouse_check_button_pressed(mb_left)

{


instance_create_layer(x,y,"Bullets",obj_bullet_ho
le);

}
```

obj_bullet_hole

Create Event:

```
image_angle=irandom(360);

alarm[0]=game_get_speed(gamespeed_fps)*2;
```

Alarm 0 Event:

```
instance_destroy();
```

45 Rope Between Objects

Draws a hanging rope between 2 positions.

Create Event:

```
for(i=0;i<2;i++)
{
    cpx[i]=random_range(30,300);
    cpy[i]=random_range(30,300);
}
mouseDragging=-1;
mouseDragX=0;
mouseDragY=0;
ropelength=300;
```

Step Event:

```
var mx=mouse_x;
var my=mouse_y;
var cx,cy;
if(mouseDragging==-1)
{
    if(mouse_check_button_pressed(mb_left))
  {
        for(i=0;i<2;i++){
            cx=cpx[i];
            cy=cpy[i];
            if(((mx-cx)*(mx-cx))+((my-cy)*(my-
cy))<400)
            {
                mouseDragging=i;
```

```
                    mouseDragX=mx-cx;
                    mouseDragY=my-cy;
                    continue;
                }
            }
        }
    }else
    {
        cpx[mouseDragging]=mx-mouseDragX;
        cpy[mouseDragging]=my-mouseDragY;

    if(mouse_check_button_released(mb_left))mouseDrag
    ging=-1;
    }


    if mouse_wheel_up()
    {
      ropelength+=10;
    }
    if mouse_wheel_down()
    {
      ropelength-=10;
    }
```

Draw Event:

```
draw_set_colour(c_blue);
for(i=0;i<2;i+
+)draw_circle(cpx[i],cpy[i],20,false);
```

```
if(point_distance(cpx[0],cpy[0],cpx[1],cpy[1])>ro
pelength)

{

    draw_set_colour(c_red);

    draw_line(cpx[0],cpy[0],cpx[1],cpy[1]);

}else

{


DrawRope(cpx[0],cpy[0],cpx[1],cpy[1],ropelength);

}
```

Script:

```
function DrawRope(xpos1, ypos1, xpos2, ypos2,
leng) {


  var x1=xpos1;

  var y1=ypos1;

  var x2=xpos2;

  var y2=ypos2;


  var xl,xr,yl,yr;

  if(xpos1>xpos2){

      xr = x1;

      yr = y1

      xl = x2;

      yl = y2

  }else{

      xl = x1;

      yl = y1

      xr = x2
```

```
        yr = y2
    }
    var l = leng;


    var s = l;
    var v = abs(y1-y2);
    var h = xr-xl;


    // check for case where just a bit apart on x-
axis - string is hanging totally vertical
    if(h<1){
        bot = (0.5*(y1+y2))+(0.5*s);
        draw_line(x1,y1,x1,bot);
        draw_line(x2,y2,x2,bot);
        exit;
    }


    // Check for string being totally taut - if so,
draw as straight line
    if((v*v)+(h*h)>=(s*s)){
        draw_line(x2,y2,x1,y1);
        exit;
    }




    var c = sqrt(s*s - v*v);
    var
h3,h5,invusq,u,hu,e,eneg,cosh,sinh,fu,dydfu,newu,
limit,oldlimit;
```

```
  // determine u (where u = 2a, and u sinh(h/u) -
c = 0 )


  // Determine good starting point for u using
Taylor series expansion

  h3 = h * h * h;

  h5 = h3 * h * h;

  invusq = (-0.166666666666666 * h3 +
sqrt(0.0277777777777777 * h3 * h3 +
0.033333333333333 * h5 * (c-h))) /
(0.01666666666666666 * h5);

  u = 1 / sqrt(invusq);


  // Improve the approximation for u using
Newton's method

  iterations = 0;

  do{

      hu = h/u;

      e = exp(hu);

      eneg = 1/e;// exp(-hu) = 1/exp(hu)

      cosh = 0.5*(e+eneg);

      sinh = 0.5*(e-eneg);

      fu = u*sinh - c;

      dydfu = sinh - hu*cosh;

      newu = abs(u - fu/dydfu); // This is where
the formula for Newton's method is applied


      limit=abs(u-newu);

      u=newu;
```

```
      iterations++;

      if(iterations>100) exit; // In case
Newton's method fails somehow, don't crash the
game

  }until(limit<0.001);


  var xv,yv;

  var a,midpoint,inva,hexp,hnexp,arcosh,midDist;


  a = 0.5*u;


  // Solve equation for known points to find x
and y offset

  // http://www.dphsw.co.uk/2017/09/29/drawing-a-
catenary/


  midPoint = 0.5*(xl+xr);

  inva = 1/a;

  if(v<1) xv=midPoint;

  else{

      hexp = exp(0.5*h*inva);

      hnexp = 1/hexp;

      sinh = 0.5*(hexp-hnexp);

      cosh = l/(a*2*sinh);

      if(cosh<=1)arcosh=0;

      else arcosh = ln(cosh+sqrt(cosh*cosh-1));

      midDist = a*arcosh;

      if(yr>yl) // In GM coords, right side is
lower
```

```
            xv = midPoint+midDist;
        else
            xv = midPoint-midDist;
    }


    e = exp((xl-xv)*inva);
    eneg = 1/e;
    yv = a * 0.5 * (e + eneg) + yl;




    // Set the starting coordinates
    var ip = xl;
    var jp = yl;


    // iterate along the rope - 20 points should be
enough to look curved
    // change to 1/40 to increase to 40 points
etc., for a higher resolution game
    var stepsize=1/20;
    var u2,u4,u8,u16,i,ix;
    for(i=stepsize;i<1;i+=stepsize){
        ix=lerp(xl,xr,i);
        u = (ix-xv)*inva;
        e=exp(u);
        eneg=1/e;
        cosh=0.5*(e+eneg);
        j = yv - a * cosh;
```

```
      // if desired, could draw a custom sprite
along

      // these coordinates instead of just a line

      // (If so, remember to also change special
cases of draw_line near top of function)

      draw_line(ip,jp,ix,j);



      ip = ix;

      jp = j;

  }


  draw_line(ip,jp,xr,yr);









}
```

46 English to Morse Code

Converts text to morse code audio.

Main conversion code:

```
text="this is an example sentence converted to
morse code"

morse_list=ds_list_create();

ds_list_add(morse_list,Pause);

source=text;

source=string_upper(source);

source_length=string_length(source);

morse="";

voice=audio_play_sound(Pause,1,false);


for (var char = 1; char < source_length+1; char
+= 1)

{

    letter="";

    letter=string_char_at(source, char);

    if letter=="A"

  {

       add=".-/";

       ds_list_add(morse_list,dot,dash,gap);

  }

  if letter=="B"

  {

       add="-.../";

       ds_list_add(morse_list,dot,dot,dot,gap);

  }
```

```
  if letter=="C"
{
     add="-.-./";
     ds_list_add(morse_list,dot,dash,dot,gap);
}
  if letter=="D"
{
     add="-../";
     ds_list_add(morse_list,dot,dot,dash,gap);
}
  if letter=="E"
{
     add="./";
     ds_list_add(morse_list,dot,gap);
}
  if letter=="F"
{
     add="..-./";
ds_list_add(morse_list,dot,dot,dash,dot,gap);
}
  if letter=="G"
{
     add="--./";
     ds_list_add(morse_list,dash,dash,dot,gap);
}
  if letter=="H"
{
```

```
      add="..../";

ds_list_add(morse_list,dot,dot,dot,dot,gap);
}
  if letter=="I"
{
      add="../";
      ds_list_add(morse_list,dot,dot,gap);
}
  if letter=="J"
{
      add=".---/";

ds_list_add(morse_list,dot,dash,dash,dash,gap);
}
  if letter=="K"
{
      add="-.-/";
      ds_list_add(morse_list,dash,dot,dash,gap);
}
  if letter=="L"
{
      add=".-../";

ds_list_add(morse_list,dot,dash,dot,dot,gap);
}
  if letter=="M"
{
      add="--/";
```

```
        ds_list_add(morse_list,dash,dash,gap);
    }
      if letter=="N"
    {
        add="-./";
        ds_list_add(morse_list,dash,dot,gap);
    }
      if letter=="O"
    {
        add="---/";
    ds_list_add(morse_list,dash,dash,dash,gap);
    }
      if letter=="P"
    {
        add=".--./";
        ds_list_add(morse_list,dash,dash,dot,gap);
    }
      if letter=="Q"
    {
        add="--.-/";
    ds_list_add(morse_list,dash,dash,dot,dash,gap);
    }
      if letter=="R"
    {
        add=".-./";
        ds_list_add(morse_list,dot,dash,dot,gap);
    }
```

```
  if letter=="S"
{
    add=".../";
    ds_list_add(morse_list,dot,dot,dot,gap);
}
  if letter=="T"
{
    add="-/";
    ds_list_add(morse_list,dash,gap);
}
  if letter=="U"
{
    add="..-/";
    ds_list_add(morse_list,dot,dot,dash,gap);
}
  if letter=="V"
{
    add="...-/";

ds_list_add(morse_list,dot,dot,dot,dot,dash,gap);
}
  if letter=="W"
{
    add=".--/";
    ds_list_add(morse_list,dot,dash,dash,gap);
}
  if letter=="X"
{
```

```
        add="-..-/";

    ds_list_add(morse_list,dash,dot,dot,dash,gap);
    }
      if letter=="Y"
    {
        add="-.--/";

    ds_list_add(morse_list,dash,dash,dot,dash,gap);
    }
      if letter=="Z"
    {
        add="--../";

    ds_list_add(morse_list,dash,dash,dot,dot,gap);
    }
      if letter==" "
    {
        add="\n";
        ds_list_add(morse_list,gap,gap);
    }


      morse+=add;
  }
```

47 Loop Through Instances

Loops through a list of instances, creating and destroying them on screen.

Create Event:

```
pos=0;

alarm[0]=game_get_speed(gamespeed_fps)*2;

list=ds_list_create();

ds_list_add(list,obj_cannon1,obj_cannon2,obj_cann
on3,obj_cannon4,obj_cannon5,obj_cannon6);

instance_create_layer(400,400,"Instances",list[|
pos]);
```

Alarm 0 Event:

```
alarm[0]=game_get_speed(gamespeed_fps)*2;

to_destroy=list[|pos];

with (to_destroy) instance_destroy();

pos++;

if pos>ds_list_size(list)-1 pos=0;

instance_create_layer(400,400,"Instances",list[|
pos]);
```

48 Slowly Rotate To Angle

A missile shooting system that targets a position.

Rotate Script:

```
function scr_rotate(startangle, targetangle, spd)

{

  return startangle +
clamp(angle_difference(targetangle,startangle), -
spd,targetangle)

}
```

Tower Step Event:

```
targetx=mouse_x; targety=mouse_y;

direction=scr_rotate(direction,point_direction(x,
y,targetx,targety),3);

image_angle=direction;
```

Missile Step Event:

```
targetx=mouse_x;

targety=mouse_y;

direction=scr_rotate(direction,point_direction(x,
y,targetx,targety),3);

image_angle=direction;

if mouse_check_button_pressed(mb_left)

{

missile=instance_create_layer(x+lengthdir_x(30,im
age_angle),y+lengthdir_y(30,image_angle),"Instanc
es",obj_missile);

  missile.image_angle=direction;

  missile.direction=direction;

  missile.speed=6;

}
```

49 Draw Clock

This draws a clock with the users system time.

Create Event:

```
minute=0;

second=0;

hour=0;

initial_angle=+90;
```

Step Event:

```
hour=current_hour;

minute=current_minute;

second=current_second;

hours=string(hour);

minutes=string(minute);

seconds=string(second);

seconds=string_repeat("0", 2-
string_length(seconds))+seconds;

minutes=string_repeat("0", 2-
string_length(minutes))+minutes;

hours=string_repeat("0", 2-string_length(hours))
+hours;
```

Draw Event:

```
//debugging draw_set_colour(c_white);

draw_set_color(c_white);

draw_text(250,50,hours+":"+minutes+":"+seconds);


//draw clock

xx=500;

yy=400;
```

```
radias=300;

draw_set_color(c_white);

draw_circle(xx,yy,radias,false);

//numbers

draw_set_colour(c_black);

draw_set_font(font_text);

draw_set_halign(fa_center);

draw_set_valign(fa_middle);

add_angle=360/12;

for (var i = 1; i < 13; i += 1)

{

    xpos=xx+lengthdir_x(radias-60,initial_angle-
(i*add_angle));

  ypos=yy+lengthdir_y(radias-60,initial_angle-
(i*add_angle));

  draw_text(xpos,ypos,i);

}

//second hand

value=360/60*second;

draw_set_colour(c_red);

handxpos=xx+lengthdir_x(radias-90,initial_angle-
value);

handypos=yy+lengthdir_y(radias-90,initial_angle-
value);

draw_line_width(xx,yy,handxpos,handypos,5);

//minue hand

value=360/60*minute;

draw_set_colour(c_blue);

handxpos=xx+lengthdir_x(radias-110,initial_angle-
value);
```

```
handypos=yy+lengthdir_y(radias-110,initial_angle-
value);

draw_line_width(xx,yy,handxpos,handypos,7);

//hour hand

value=360/12*hour;

mins=12/60*minute*2

value=value+mins;

draw_set_colour(c_black);

handxpos=xx+lengthdir_x(radias-110,initial_angle-
value);

handypos=yy+lengthdir_y(radias-110,initial_angle-
value);

draw_line_width(xx,yy,handxpos,handypos,9);
```

50 Randomly Place Objects In Room

Randomly places instances in the room.

Main spawn code in **Create Event**:

```
list=ds_list_create();

ds_list_add(list,obj_cannon1,obj_cannon2,obj_cann
on3,obj_cannon4,obj_cannon5,obj_cannon6);

for (var i = 0; i < ds_list_size(list); i += 1)

{


instance_create_layer(irandom_range(100,room_widt
h-100),irandom_range(100,room_height-
100),"Instances",list[|i]);

}
```

51 Get Text From Keyboard

Allows user to enter text using the keyboard.

Create Event:

```
keyboard_string="";

has_done=false;

done="";
```

Step Event:

```
if mouse_check_button_pressed(mb_left)

{

  game_restart();

}

if has_done exit;

if keyboard_check_pressed(vk_enter)

{

  done=keyboard_string;

  has_done=true;

}
```

Draw Event:

```
draw_set_font(font_text);

draw_set_colour(c_white);

draw_text(50,50,"Type Using Keyboard - Enter To
Set - Delete To Clear \nLeft Mouse Button
Restart");

if !has_done draw_text(50,150,"Typed Text:
"+keyboard_string);

if has_done

{

  draw_text(50,250,"Entered "+string(done)); }
```

52 Shoot Projectile With Gravity

Allows player to shoot a projectile, with gravity applied.

Tower Step Event:

```
image_angle=point_direction(x,y,mouse
_x,mouse_y);

if
mouse_check_button_pressed(mb_left)

{

  xx=lengthdir_x(140,image_angle);

  yy=lengthdir_y(140,image_angle);

  instance_create_layer(x+xx, y+yy,
"Instances", obj_bullet,

  {

      speed : 10,

      direction : image_angle,

      image_angle : image_angle,

      gravity : 0.2

  });

}
```

Bullet Step Event:

```
image_angle=direction;

if y>room_height instance_destroy();
```

53 Fade On Player Collision

This fades and destroys an instance if the player collides with it.

Create Event:

fading=false;

alp=1;

Step Event:

```
if fading=true

{

  alp-=0.01;

}

if alp<=0 instance_destroy();
```

Draw Event:

```
draw_sprite_ext(sprite_index,0,x,y,1,1,0,c_white,
alp);
```

Set `fading` to `true` when you want it to be destroyed.

54 Jump On Enemy To Kill

Allows player to jump on enemy to destroy it.

Enemy Step Event:

```
inst=instance_place(x,y+2,obj_player);
if inst!=noone
{ if inst.bbox_bottom>bbox_top && inst.vspeed>0.1
  instance_destroy();
}
```

55 Calculate Size Of Area

Allows user too click to positions on screen and calculate the size:

Create Event:

```
click=0;

mess="Click Corner Of Rectangle";
```

Step Event:

```
if click==2

{

width=x2-x1;

height=y2-y1;

width=abs(width);

height=abs(height);

click++;

}

if mouse_check_button_released(mb_left) &&
click==1

{

x2=mouse_x;

y2 =mouse_y;

click+=1;

mess="Right Click To Restart";

}

if mouse_check_button_released(mb_left) &&
click==0

{

x1=mouse_x;
```

```
y1= mouse_y;

click+=1;

mess="Click Second Corner Of Rectangle";

}


if mouse_check_button_pressed(mb_right) &&
click==3

{

  room_restart();

}
```

Draw Event:

```
draw_set_colour(c_black);

draw_set_font(font_text);

if click==3

{

  sides_text=string(2*(width+height));

  area_text=string(width*height);

  height_text=string(height);

  width_text=string(width);

  draw_set_colour(c_blue);

    draw_rectangle(x1,y1,x2,y2,false);

    draw_set_colour(c_black);

    draw_text(10,40,"width "+width_text);

    draw_text(10,80,"height "+height_text);

    draw_text(10,120,"perimeter "+sides_text);

    draw_text(10,160,"area "+area_text);


}
```

```
draw_set_colour(c_black);
draw_set_font(font_text);
draw_text(10,700,mess);
```

56 Draw Lines To Mouse Position

Draw coloured line between two positions.

Create Event:

```
target_x=0;

target_y=0;

start_x=50;

start_y=room_height-50;
```

Step Event:

```
target_x=0;

target_y=0;

start_x=50;

start_y=room_height-50;

ang=0;
```

Draw Event:

```
draw_set_colour(c_white);

draw_set_font(font_text);

draw_text(50,50,"Move mouse");

draw_set_colour(c_blue);

draw_line_width(start_x,start_y,target_x,target_y
,9);

draw_set_colour(c_yellow);

draw_line_width(start_x,start_y,target_x,target_y
,3);
```

57 Random Building Generator

Generates buildings of random height and colour.

Create Event:

```
grid_size=32;

size=floor(room_width/grid_size);

global.level=2;

for(loop=1;loop<size+1;loop+=1)

{


height_of_building=irandom_range(2,10)+global.lev
el;

  building_array[loop,0]=height_of_building;


colour_of_building=choose(c_silver,c_olive,c_fuch
sia,c_aqua,c_lime,c_red,c_green,c_white,c_yellow,
c_purple,c_orange);

  building_array[loop,1]=colour_of_building;

}



for(loop=1;loop<31;loop+=1)

{

  xpos=grid_size*loop;


for(height=1;height<building_array[loop,0];height
+=1)

  {

       ypos=room_height-(grid_size*height);



building=instance_create_layer(xpos,ypos,"Instanc
es",obj_block);
```

```
  building.image_blend=building_array[loop,1]
       building.image_index=irandom_range(0,2);
       building.image_speed=0;
  }
  building=instance_create_layer(xpos,ypos-
32,"Instances",obj_block);
  building.image_blend=building_array[loop,1]
  building.image_index=3;
  building.image_speed=0;
}
```

Also uses **obj_block** with sprites assigned.

58 Queue & Play Audio

Queues audio. Great for queuing message and sound effects.

Create Event:

```
list=ds_list_create();

ds_list_add(list,Accessible_Gaming,Pause,Avoid_Th
e_Enemies,Pause,Use_Left_Mouse_Button_To_Move,Pau
se,You_Lost_This_Round)

voice=audio_play_sound(Pause,1,false);
```

Step Event:

```
if !audio_is_playing(voice)

{

  if ds_list_size(list)!=0

  {

       to_play=list[|0];

       voice=audio_play_sound(to_play,1,0);

       ds_list_delete(list,0);

  }



}



if mouse_check_button_pressed(mb_left)

{

  room_restart();

}
```

Draw Event:

```
draw_set_font(c_white);
draw_set_font(font_text);
size=ds_list_size(list);
draw_text(100,200,size);
```

59 Boss Style Movement

A boss style object with a repeating movement and shooting pattern.

Create Event:

```
alarm[0]=game_get_speed(gamespeed_fps)/2;
```

Step Event:

```
if y<starty-60 && has_fired==false
{
  alarm[0]=game_get_speed(gamespeed_fps)/2;
  has_fired=true;
}
```

Alarm 0 Event:

```
bullet_count++;
if bullet_count<4
{
  alarm[0]=game_get_speed(gamespeed_fps)/2;

instance_create_layer(x,y,"Instances",obj_bullet,
  {
       hspeed : -3
  })
}
```

Path Ended Event:

```
has_fired=false;
path_start(Path1,6,path_action_stop,true);
bullet_count=0;
```

60 Split Screen

This example tracks two instances.

There is no code as this is done in the room’s views settings.

See example project file.

61 Check Spelling Of Word

Allows you to check if a word exists within an included dictionary text file. Great for word themed games.

Basic example **Create Event:**

```
dictionary=ds_list_create();

dic_file=file_text_open_read("dictionary.txt");

while(!file_text_eof(dic_file))

{


    word=file_text_read_string(dic_file);

    ds_list_add(dictionary,word);
```

To check:

```
word_to_find=string;

  position = ds_list_find_index(dictionary,
word_to_find);

  if position>0

  {

       result="Is In Dictionary";

  }

  else

  {

      result="Not In Dictionary";

  }
```

62 Player Character Selection

Allows player to choose their character that will be their sprite in game.

Create Event:

```
current=1;
alarm[0]=game_get_speed(gamespeed_fps)*2;
sprite_index=spr_char_1;
```

Step Event:

```
if mouse_check_button(mb_left)
{
  global.character=sprite_index;
  room_goto(room_game);
}
```

Alarm 0 Event:

```
alarm[0]=game_get_speed(gamespeed_fps)*2;
current++;
if current=6 current=1;
if current==1 sprite_index=spr_char_1;
if current==2 sprite_index=spr_char_2;
if current==3 sprite_index=spr_char_3;
if current==4 sprite_index=spr_char_4;
if current==5 sprite_index=spr_char_5;
```

Draw Event:

```
x=100;
y=100;
draw_self();
```

To set the character, use something like the following in the **Create Event** of your character:

```
sprite_index=global.character;
```

63 Weapon Control & Ammo Packs

This keeps the instance within the room's boundary. Assumes sprite origin as center.

Create Event:

```
ammo=6;
```

Step Event:

```
if mouse_check_button_pressed(mb_left)
{
  if ammo>0
  {
       audio_play_sound(snd_gun,1,0);
       ammo--;
  }
  else
  {
       audio_play_sound(no_ammo,1,false);
  }
}
```

Draw Event:

```
draw_sprite(spr_bullets_left,ammo,100,100);
```

To Add Ammo:

```
obj_example.ammo=6;
audio_play_sound(extra_ammo,1,false);
```

64 Move Towards Point Then Stop

Moves an instance towards a point and then stops.

Create Event:

```
targetx=x;
targety=y;
spd=5;
```

Step Event:

```
if mouse_check_button_pressed(mb_left)
{
  targetx=floor(mouse_x);
  targety=floor(mouse_y);
}
dist=point_distance(x,y,targetx,targety);
if dist>0
{
  if dist>spd
  {
       move_towards_point(targetx,targety,spd);
  }
  else
  {
       move_towards_point(targetx,targety,1);
  }

}
```

65 Resize Based On Position

Scales an instance based on it’s Y position.

Create Event:

```
scale=0;
```

Step Event:

```
y=mouse_y;
y=clamp(y,128,room_height-128);
var ymin = 128,
    ymax = 640,
    range = ymax - ymin;
scale = (y - ymin) / range;
image_xscale=scale;
image_yscale=scale;
```

66 Using Mouse Wheel To Select Weapon

Allows user to change weapon using middle mouse wheel.

Create Event:

```
weapon=0;
```

Step Event:

```
if mouse_wheel_up()
{
  weapon++;
}
if mouse_wheel_down()
{
  weapon--;
}
if weapon=image_number
{
  weapon=0;
}
if weapon=-1
{
  weapon=image_number-1;
}
```

Draw Event:

```
draw_sprite(sprite_index,weapon,x,y);

draw_text(100,100,string(weapon+1)+" of
"+string(image_number));
```

67 Font Drawing From Images

Example for drawing sprites created from images.

Create Event:

```
global.font_example=font_add_sprite_ext(spr_font,
"1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZ ",false,4);

example="Hello World Example 1267";

example=string_upper(example);
```

Draw Event:

```
draw_set_font(global.font_example);

draw_text(100,100,example);
```

68 Allow Player To Load Sprite

Allows player to load an image file from their computer. Great to add a player's avatar.

Create Event:

```
var image;

image=get_open_filename("Image File|*.png", "");

spr_loaded=sprite_add(image,0,true,false,0,0);
```

Draw Event:

```
draw_sprite(spr_loaded,0,200,200);
```

69 Enemy Shoots If Can See Player

This system make the enemy shoot a bullet if it can see the player.

Create Event:

```
can_see=false;

alarm[0]=game_get_speed(gamespeed_fps)*2;
```

Step Event:

```
if
collision_line(x,y,obj_player.x,obj_player.y,obj_
crate, false, false)

{

  can_see=false;

}

else

{

  can_see=true;

}

if can_see

{


move_towards_point(obj_player.x,obj_player.y,1);


angle_to_player=point_direction(x,y,obj_player.x,
obj_player.y);


ang=angle_difference(image_angle,angle_to_player)
;

  if ang>0 image_angle--;

  else

  image_angle++;
```

```
}
else
{
  speed=0;
}
```

Alarm 0 Event:

```
alarm[0]=game_get_speed(gamespeed_fps)*2;
if can_see
{
  instance_create_layer(x,y,"Bullets",obj_bullet,
  {
    speed : 5,
    direction : image_angle,
  image_angle : image_angle
  })
}
```

70 Randomly Place Instances Avoiding Instances

This spawns a number of instances, no closer than a given distance to other defined instances.

Create Event:

```
repeat 20
{
  xx=irandom(room_width);
  yy=irandom(room_height);
  inst=instance_nearest(xx,yy,obj_crate);
  dist=point_distance(xx,yy,inst.x,inst.y);


  while dist<200
  {
       xx=irandom(room_width);
       yy=irandom(room_height);
       inst=instance_nearest(xx,yy,obj_crate);
       dist=point_distance(xx,yy,inst.x,inst.y);
  }

instance_create_layer(xx,yy,"Instances",obj_gem);
}
```

71 Split Sentence

This will split a sentence in to multiple lines of text, with breaks at end of words.

Script:

```
function split_text_into_lines(text, maxWidth,
font) {

    var lines = [];

    var currentLine = "";

    var words = string_split(text, " ");



    draw_set_font(font);



    for (var i = 0; i < array_length(words); i++)
{

        var word = words[i];



        if (string_width(currentLine + " " +
word) <= maxWidth) {

            currentLine = currentLine +
(currentLine == "" ? "" : " ") + word;

        } else {

            lines[array_length(lines)] =
currentLine;

            currentLine = word;

        }

    }



    if (currentLine != "") {

        lines[array_length(lines)] = currentLine;
```

```
    }


    return lines;

}
```

Example usage.

Create Event:

```
var text="Lorem ipsum dolor sit amet, consectetur
adipiscing elit, sed do eiusmod tempor incididunt
ut labore et dolore magna aliqua. Ut enim ad
minim veniam, quis nostrud exercitation ullamco
laboris nisi ut aliquip ex ea commodo consequat.
Duis aute irure dolor in reprehenderit in
voluptate velit esse cillum dolore eu fugiat
nulla pariatur. Excepteur sint occaecat cupidatat
non proident, sunt in culpa qui officia deserunt
mollit anim id est laborum.";

var max_length=1000;

var font=font_text;

text_array =
split_text_into_lines(text,max_length,font);
```

Draw Event:

```
draw_set_font(font_text);

for (var i = 0; i < array_length(text_array); i+
+)

{

    draw_text(100,200+(i*45),text_array[i]);

}
```

72 Simple Menu System

A simple adaptable menu system.

Create Event:

```
current=1;
alarm[0]=game_get_speed(gamespeed_fps)*4;
button_text[1]="Play Game";
button_text[2]="Level Select";
button_text[3]="Quit Game";
```

Alarm 0 Event:

```
alarm[0]=game_get_speed(gamespeed_fps)*4;
current++;
if current=4 current=1;
```

Step Event:

```
if mouse_check_button_pressed(mb_left)
{
  if current=1 room_goto(room_play);
  if current=2 room_goto(room_lev_select);
  if current=3 room_goto(room_exit);
}
```

Draw Event:

```
draw_set_font(font_button);
draw_set_halign(fa_center);
draw_set_valign(fa_middle);
xpos=room_width/2;
```

```
ypos=200;
if current=1
{
  draw_sprite(spr_button,0,xpos,ypos);
}
else
{
  draw_sprite(spr_button,1,xpos,ypos);
}
draw_text(xpos,ypos,button_text[1]);

ypos=400;
if current=2
{
  draw_sprite(spr_button,0,xpos,ypos);
}
else
{
  draw_sprite(spr_button,1,xpos,ypos);
}
draw_text(xpos,ypos,button_text[2]);

ypos=600
if current=3
{
  draw_sprite(spr_button,0,xpos,ypos);
}
else
```

```
{
  draw_sprite(spr_button,1,xpos,ypos);
}
draw_text(xpos,ypos,button_text[3]);
```

73 Moving Spikes & Damage System

Creates a spike that fades in and out, giving player damage with the alpha is greater than 0.5.

Create Event of **obj_spike**:

```
alp=1;

fade_speed=0.01;

out=true;
```

Step Event:

```
if out==true

{

  alp-=fade_speed;

  if alp<-1

  {

       out=false;

  }

}

else

{

  alp+=fade_speed;

  if alp>1

  {

       out=true;

  }

}
```

Draw Event:

```
draw_sprite_ext(sprite_index,0,x,y,1,1,0,c_white,
alp);
```

Example collision code called from **obj_player Collision Event** with spike object:

```
if other.alp>0.5
{
  if !audio_is_playing(Arrghh)
  {
       audio_play_sound(Arrghh,1,false);
  }
}
```

74 Projectile Spread System

Shoot multiple projects at slightly different angle.

Obj Tower Step Event:

```
targetx=mouse_x;

targety=mouse_y;

direction=point_direction(x,y,targetx,targety)

image_angle=direction;
```

Step Event:

```
targetx=mouse_x;

targety=mouse_y;

direction=point_direction(x,y,targetx,targety)

image_angle=direction;

if mouse_check_button_pressed(mb_left)

{


missile=instance_create_layer(x+lengthdir_x(30,im
age_angle),y+lengthdir_y(30,image_angle),"Instanc
es",obj_missile);

  missile.image_angle=direction;

  missile.direction=direction;

  missile.speed=6;


missile=instance_create_layer(x+lengthdir_x(30,im
age_angle),y+lengthdir_y(30,image_angle),"Instanc
es",obj_missile);

  missile.image_angle=direction-20;

  missile.direction=direction-20;

  missile.speed=6;
```

```
missile=instance_create_layer(x+lengthdir_x(30,im
age_angle),y+lengthdir_y(30,image_angle),"Instanc
es",obj_missile);

  missile.image_angle=direction+20;

  missile.direction=direction+20;

  missile.speed=6;

}
```

obj_missile Outside Room Event:

```
instance_destroy();
```

75 Ball Bounce & Squash

Makes a ball bounce with a squashing effect.

Create Event:

```
gravity=0.1;

dir="down"

has_hit=false;

vspeed-=5;
```

Step Event:

```
if position_meeting(x,y+1,obj_crate) &&
has_hit=false

{

  gravity=0;

  vspeed=0;

  has_hit=true;

}

if has_hit && dir="down"

{

  image_yscale-=0.01;

  if image_yscale<0.6

  {

       dir="up";

  }

}

if has_hit && dir="up"

{

  image_yscale+=0.01;

  if image_yscale>=1
```

```
    {
        vspeed=-5;
        has_hit=false;
        gravity=0.1;
        dir="down";
    }
}
```

76 Status Effect

Draws info on what the player is doing.

Create Event:

```
mess="Up";
```

Step Event:

```
if (keyboard_check(ord("W")))
{
  sprite_index=spr_up;
  image_speed=1;
  mess="Up";
}
else if (keyboard_check(ord("S")))
{
  sprite_index=spr_down;
  image_speed=1;
  mess="Down";
}

else
if (keyboard_check(ord("A")))
{
    sprite_index=spr_left;
  image_speed=1;
       mess="Left";
}
else
if (keyboard_check(ord("D")))
```

```
{
    sprite_index=spr_right;
  image_speed=1;
       mess="Right";
}
else
{
  image_speed=0;
       mess="Idle";
}
```

Draw Event:

```
draw_self();
draw_text(500,100,"Player is "+mess);
```

77 Foot Step Sounds With Animation

A simple method to tie animation and sound effects.

Step Event:

```
if image_index=0 or image_index=12
{
  audio_play_sound(snd_footstep,1,false);
}
```

78 Game Fog

Draws a fog effect over the room using a sprite.

Create Event:

```
fog=true;
```

Step Event:

```
if mouse_check_button_pressed(mb_left)
{
  fog=!fog;
}
```

Draw Event:

```
if fog
{

draw_sprite_ext(spr_fog,0,0,0,1,1,0,c_white,0.6)
}
```

79 Destruction With Multiple Subimages

A destrucable crate with multiple subimages that is destroyed when all image frames have been used.

Crate Event:

```
image_speed=0;

image_index=0;

sub=0;

total=image_number-1;
```

Step Event:

```
if mouse_check_button_pressed(mb_left)

{

  if instance_position(mouse_x,mouse_y,id)

  {

        sub++;

        image_index=sub;

  }

}

if sub>total-1 instance_destroy();
```

80 Enemy Hide

Makes an enemy find a hiding place if the player can see it.

Create Event:

```
can_see=false;

moving=false;

xx=0;

yy=0;

found=false;

path=path_add();

grid = mp_grid_create(0,0,
room_width/32,room_height/32,32,32);

mp_grid_add_instances(grid,obj_crate,true);
```

Step Event:

```
if can_see && found=false

{

  var xx, yy;

    do

  {

        xx = irandom(room_width);

        yy = irandom(room_height);

    } until (!instance_position(xx, yy,
obj_crate) && collision_line(xx, yy,
obj_player.x, obj_player.y, obj_crate, false,
false) != noone)

    found=true;

  mp_grid_path(path, grid, x, y, xx, yy, true);

    path_start(path, 5, path_action_stop, true);

}
```

Path End Event:

```
found=false;

can_see=false;
```

81 HUD Drawing On GUI Layer

Draws a HUD on the GUI layer so it been seen when the view moves.

Create Event:

```
lives=5;

score=734;

hp=100;
```

Step Event:

```
if keyboard_check(ord("Z")) hp--;

if keyboard_check(ord("X")) hp++;

hp=clamp(hp,0,100);
```

Draw Event:

```
var xsize=camera_get_view_width(view_camera[0]);

var ysize=camera_get_view_height(view_camera[0]);

draw_set_colour(c_blue);

draw_set_alpha(0.5);

draw_roundrect(30,30,xsize-30,80,false);

draw_set_alpha(1);

draw_set_colour(c_white);

draw_roundrect(20,20,xsize-20,ysize-20,true);

draw_roundrect(30,30,xsize-30,80,true);

draw_set_font(font_text);

draw_set_halign(fa_center);

draw_set_valign(fa_middle);

draw_text(120,55,"LIVES");

for (var i=0; i<lives; i+= 1)

{
```

```
  draw_sprite(spr_lives,0,220+(55*i),55)
}
draw_set_halign(fa_right);
var str = string(score);
draw_text(xsize-50, 55,"SCORE
"+string_repeat("0", 6-string_length(str))+str);
scale = relerp(0, 100, hp, 0, 1);
draw_sprite_ext(spr_bar_bg,0,xsize/2-
92,55,scale,1,0,c_white,1);
draw_sprite(spr_bar,0,xsize/2,55);
```

82 Scroll Block Of Text Up and Down

This allows for scrolling through a large block of text.

Create Event:

```
var text="Lorem ipsum dolor sit amet, consectetur
adipiscing elit, sed do eiusmod tempor incididunt
ut labore et dolore magna aliqua. Ut enim ad
minim veniam, quis nostrud exercitation ullamco
laboris nisi ut aliquip ex ea commodo consequat.
Duis aute irure dolor in reprehenderit in
voluptate velit esse cillum dolore eu fugiat
nulla pariatur. Excepteur sint occaecat cupidatat
non proident, sunt in culpa qui officia deserunt
mollit anim id est laborum. Lorem ipsum dolor sit
amet, consectetur adipiscing elit, sed do eiusmod
tempor incididunt ut labore et dolore magna
aliqua. Ut enim ad minim veniam, quis nostrud
exercitation ullamco laboris nisi ut aliquip ex
ea commodo consequat. Duis aute irure dolor in
reprehenderit in voluptate velit esse cillum
dolore eu fugiat nulla pariatur. Excepteur sint
occaecat cupidatat non proident, sunt in culpa
qui officia deserunt mollit anim id est
laborum.";

var max_length=1000;

var font=font_text;

text_array =
split_text_into_lines(text,max_length,font);

ypos=0;

height=0;
```

Step Event:

```
if keyboard_check(ord("W"))

{

  ypos--;

}

if keyboard_check(ord("S"))
```

```
{
  ypos++;
}
min_lines=8;
height=(array_length(text_array)-min_lines)*45;
ypos=clamp(ypos,0,height);
```

Draw Event:

```
draw_set_font(font_text);
for (var i = 0; i < array_length(text_array); i++)
{
    draw_text(50,(-ypos)+(i*45),text_array[i]);
}
draw_text(1080,50,"WS To Move\nText Up and Down");
```

83 Blood Spray Effect

An adaptable effect for creating a blood style explosion.

Create Event:

```
count=0;

direction=irandom_range(50,140)

image_angle=direction;

speed=random(5);

scale=random_range(0.1,1);

alp=1;

fade_speed=random_range(0.001,0.02);

image_xscale=scale;

image_yscale=scale;

image_index=choose(0,1);
```

Step Event:

```
alp-=fade_speed;

if alp<=0 instance_destroy();
```

Draw Event:

```
draw_sprite_ext(sprite_index,image_index,x,y,imag
e_xscale,image_yscale,image_angle,c_white,alp);
```

Example spawn code in **control object**:

```
if mouse_check_button_pressed(mb_left)

{repeat(30)

  {

instance_create_layer(mouse_x,mouse_y,"Instances"
,obj_blood);

  }}
```

84 Voice On Level Up

Level up system that plays a voice on level up or level down.

Create Event:

```
boost=0;

level=1;

previous=1;

audio_play_sound(lev_1,1,false);
```

Step Event:

```
if mouse_check_button_pressed(mb_left)

{

  boost--;

}

if mouse_check_button_pressed(mb_right)

{

  boost++;

}


boost=clamp(boost,0,60);

if boost>60 boost=60;

if boost>0 and boost<10

{

  level=1;

  if previous!=level

  {

  audio_play_sound(lev_1,1,false);

       previous=1;
```

```
    }
}
if boost>10 and boost<20
{
    level=2;
    if previous!=level
    {
        audio_play_sound(lev_2,1,false);
        previous=2;
    }
}
if boost>20 and boost<30
{
    level=3;
    if previous!=level
    {
        audio_play_sound(lev_3,1,false);
        previous=3;
    }
}
if boost>30 and boost<40
{
    level=4;
    if previous!=level
    {
        audio_play_sound(lev_4,1,false);
        previous=4;
    }
```

```
}
if boost>40 and boost<50
{
  level=5;
  if previous!=level
  {
       audio_play_sound(lev_5,1,false);
       previous=5;

  }
}

if boost>50 and boost<60
{
  level=6;
  if previous!=level
  {
       audio_play_sound(lev_6,1,false);
       previous=6;
  }
}
```

Draw Event:

```
draw_set_font(font_info);
draw_set_colour(c_white);
draw_text(50,50,"Left and Right Mouse Buttons To
Change Value");
draw_text(50,100,"Value "+string(boost));
```

85 Wind Blown Effect

Makes a sprite move like it has a wind force on it,

Script:

```
function
draw_grass_with_wind(sprite,subimg,x,y,amplitude,
frequency,phase){

var w=sprite_get_width(sprite);

var h=sprite_get_height(sprite)/2;

var wave1=amplitude*sin(frequency*0+phase);

var wave2=amplitude*sin(frequency*0.5*h+phase);

var wave3=amplitude*sin(frequency*h+phase);

var x1=x+wave1;

var y1=y;

var x2=x+w+wave1;

var y2=y;

var x3=x+wave2;

var y3=y+h*0.5;

var x4=x+w+wave2;

var y4=y+h*0.5;

var x5=x+wave3;

var y5=y+h;

var x6=x+w+wave3;

var y6=y+h;

draw_primitive_begin_texture(pr_trianglestrip,spr
ite_get_texture(sprite,subimg));

draw_vertex_texture(x1,y1,0,0);

draw_vertex_texture(x2,y2,1,0);

draw_vertex_texture(x3,y3,0,0.5);
```

```
draw_vertex_texture(x4,y4,1,0.5);

draw_vertex_texture(x5,y5,0,1);

draw_vertex_texture(x6,y6,1,1);

draw_primitive_end();

}
```

Draw Event:

```
sprite_index=spr_grass;

var subimg_index=0;

var x_position=100;

var y_position=100;

var  amplitude=10;

var frequency=0.05;

var phase=current_time/1000;

draw_grass_with_wind(sprite_index,subimg_index,x_
position,y_position,amplitude,frequency,phase);
```

86 Double Jump

Allow player to perform a double jump.

Create Event:

```
gravity=0.1;
jumps=0;
```

Step Event:

```
if instance_place(x, bbox_bottom + 1, obj_crate)
{
    vspeed=0;
  y=y-1;
  jumps=0;
}
if mouse_check_button_pressed(mb_left) && jumps<2
{
  vspeed=-6;
  jumps++;
}
```

87 Meteor Shower Effect

Creates a meteor and explosion effect, great to add atmosphere to your game.

Spawn object Step Event example:

```
if mouse_check_button_pressed(mb_left)
{
  repeat(6)
  {
       xpos=irandom_range(50,room_width-50);
       instance_create_layer(xpos,-
100,"Instances",obj_meteor);
  }
}
```

obj_meteor Create Event:

```
count=0;
direction=irandom_range(220,320);
image_angle=direction;
speed=5;
destroy_at=irandom_range(50,170);
```

Step Event:

```
count++;
if count>destroy_at
{
  repeat(10)
  {
```

```
instance_create_layer(x,y,"Instances",obj_explosi
on);

  }

  instance_destroy();

}
```

obj_explosion Create Event:

```
image_angle=irandom(359);

direction=image_angle;

speed=5;
```

Animation End Event:

```
instance_destroy();
```

88 Footstep Dust Effect

Makes a small effect when an instance walks.

Just pop this code in the Step Event:

```
if floor(image_index)=7 or floor(image_index)=18
{
  effect_create_above(ef_smoke,x,y,3,c_maroon);
}
```

89 Float & Die Effect

Shows an enemy dead and makes it wobble and fade.

Create Event:

```
alive=true;
hp=100;
alp=1;
ang=0;
sw=2;
```

Step Event:

```
if alive
{
  hp--;
}
if hp<=0
{
  alive=false;
  image_speed=0;
}
if !alive
{
  alp-=0.004;
  y--;
  sw+=0.2;
  ang=sin(sw)*5;
}
if alp<0 instance_destroy();
```

Draw Event:

```
if alive

{

  draw_self();

}

else

{


draw_sprite_ext(sprite_index,image_index,x,y,1,-
1,ang,c_white,alp);

}

draw_set_colour(c_white);

draw_text(x,y-100,hp);
```

90 Fly Level Effect

Return a flying plane to middle when player stops movement.

Step Event:

```
if keyboard_check(ord("W"))
{
  y-=2;
}
else
if keyboard_check(ord("S"))
{
  y+=2;
}
else
{
  if y>room_height/2
  {
       y--;
  }
  if y<room_height/2
  {
       y++;
  }
}
y=clamp(y,50,room_height-50);
```

91 Dash Movement

Allows player to dash in the direction it is facing.

Create Event:

```
on_ice=false;
dash=false;
count=0;
count_max=30;
```

Step Event:

```
if keyboard_check(ord("A"))
{
  x-=2;
  image_xscale=-1;
}
if keyboard_check(ord("D"))
{
  x+=2;
  image_xscale=1;
}

if keyboard_check_pressed(vk_space)
{
  dash=true;

}
if dash
{
```

```
  x+=sign(image_xscale)*6
  count++;
}
if count>=count_max
{
  dash=false;
  count=0;
}
```

92 Walking On Ice

Makes a player slide when moving on ice.

Create Event:

```
on_ice=false;
```

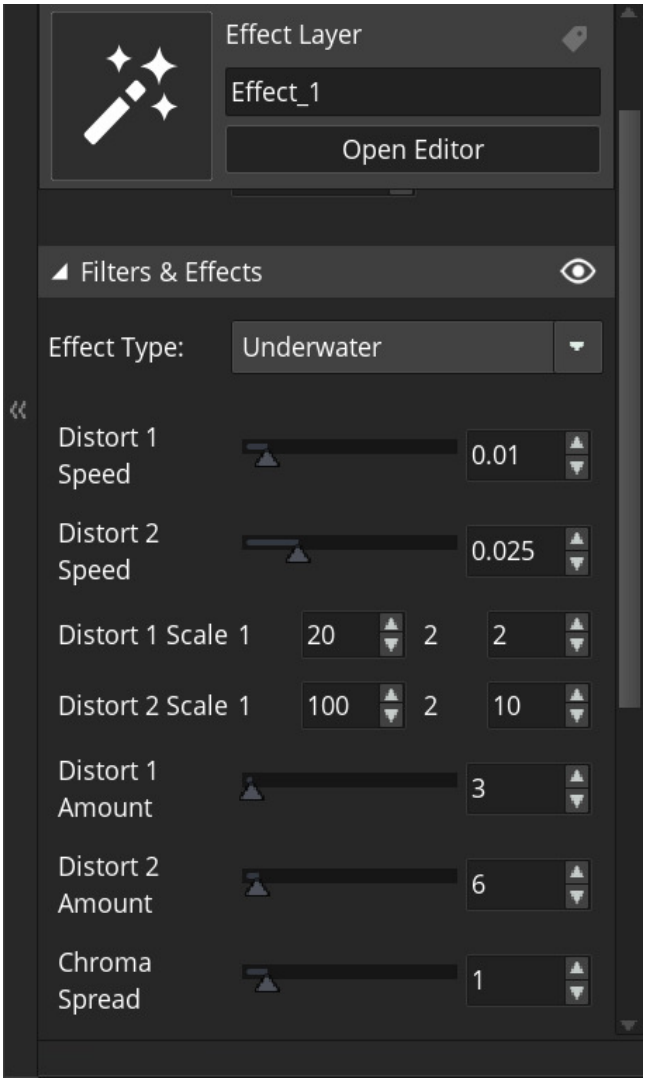
Step Event:

```
if position_meeting(x,y+2,obj_ice)
{
  on_ice=true;
}
else
{
  on_ice=false;
}
if on_ice
{
  if keyboard_check(ord("A"))
  {
       motion_add(180,.2);
  }
  else if keyboard_check(ord("D"))
  {
       motion_add(0,.2);
  }
  else
  {
       hspeed=hspeed*0.99;
  }
```

```
}
if !on_ice
{
  if keyboard_check(ord("A"))
  {
       x-=2;
  }
  else if keyboard_check(ord("D"))
  {
       x+=2;
  }
  else
  {
       hspeed=0;
  }
}
x=clamp(x,sprite_width,room_width-sprite_width);
```

93 Underwater Effect

Creates an underwater effect using effects layer and sprites.



94 Hint Arrow To Direction Of Powerup

Gives the player a hint of which direction the nearest power up is.

Create Event:

```
target=noone;

ang=0;

target_ang=0;

diff=0;
```

Step Event:

```
x=mouse_x;

y=mouse_y;

if instance_exists(obj_powerup)

{

  target=instance_nearest(x,y,obj_powerup);


target_ang=point_direction(x,y,target.x,target.y)
;

  diff=angle_difference(ang,target_ang);

  if sign(diff)>0 ang-- else ang++

}

else

{

  target=noone;

}
```

Draw Event:

```
draw_self();

if instance_exists(target)

{


draw_sprite_ext(spr_arrow,0,x,y,1,1,ang,c_white,1
);

}
```

95 Button To Open Website

A button that the player can click to visit a website.

Step Event for an object with a button assigned:

```
if mouse_check_button_pressed(mb_left) &&
instance_position(mouse_x,mouse_y,id)

{

  url_open("http://gamemakerexamples.com");

}
```

96 Health Pack Slowly Increase Health

Slowly increases the player's health when they collect a health pack.

Create Event:

```
health=20;

target=20;
```

Step Event:

```
inst=noone

if mouse_check_button_pressed(mb_left)

{


inst=instance_position(mouse_x,mouse_y,obj_health
);

  if inst!=noone

  {

        target=target+20;

        with inst instance_destroy();

  }

}


if health<target

{

  health+=0.2;

  show_debug_message(health);

}
```

Draw Event:

```
draw_healthbar(100,300,500,350,health,c_grey,c_re
d,c_green,0,true,true);
```

97 Change Enemy Colour When Targeted

Changes the enemy colour when it is targeted.

Controller Create Event:

```
global.selected=noone;
```

Step Event:

```
if mouse_check_button_pressed(mb_left)
{
  var list=array_create();
  with obj_enemy
  {
       array_push(list,id);
  }
  var size=array_length(list);
  {
       choice=irandom(size-1);
       global.selected=list[choice];
  }
  array_delete(list);
}
```

obj_enemy Draw Event:

```
if global.selected=id

{


draw_sprite_ext(sprite_index,0,x,y,1,1,0,c_red,1)
;

}

else

{

  draw_self();

}
```

98 Limit Weapon Shooting Timer

Limits how quickly the player can shoot their weapon.

Create Event:

```
delay=game_get_speed(gamespeed_fps)*3;

can_shoot=true;
```

Step Event:

```
if mouse_check_button_pressed(mb_left) &&
can_shoot

{

  can_shoot=false;

  alarm[0]=delay;

  instance_create_layer(x,y,"bullet",obj_missile,

  {

       hspeed : 5});

}
```

Alarm 0 Event:

```
can_shoot=true;
```

Draw Event:

```
draw_self();

if can_shoot

{

  draw_text(x,y-180,"Can Shoot");

}

else

{

  draw_text(x,y-180,"Wait");

}
```

99 Clock Stopwatch

A stop watch time with hundredths of seconds.

Create Event:

```
stopwatch_time=0;
stopwatch_running=false;
game_set_speed(100,gamespeed_fps);
```

Step Event:

```
if (stopwatch_running)
{
    stopwatch_time+=1;
}
if mouse_check_button_pressed(mb_left)
{
  stopwatch_running=!stopwatch_running;
}
if mouse_check_button_pressed(mb_right)
{
  stopwatch_running=false;
  stopwatch_time=0;
}
```

Draw Event:

```
var total_seconds=floor(stopwatch_time/100);
var minutes=floor(total_seconds/60);
var seconds=total_seconds mod 60;
var hundredths=stopwatch_time mod 100;
minutes=string(minutes);
```

```
minutes=string_repeat("0",2-
string_length(minutes))+minutes;

seconds=string(seconds);

seconds=string_repeat("0",2-
string_length(seconds))+seconds;

hundredths=string(hundredths);

hundredths=string_repeat("0",2-
string_length(hundredths))+hundredths;

draw_text(100,110,minutes+":"+seconds+":"+string(
hundredths));
```

100 Weapon Power & Direction System

Allows user to choose an angle and power of a projectile.

Create Event:

```
ang=45;

dir=1;

pow=0;

state="aim";
```

Step Event:

```
if state="aim"

{

  ang+=dir;

  if ang>90 dir=-1;

  if ang<20 dir=+1;

}

if state="power"

{

  pow+=dir;

  if pow>100 dir=-1;

  if pow<20 dir=+1;

}

image_angle=ang;

if mouse_check_button_pressed(mb_left) &&
state="aim"

{

  state="power";

  exit;
```

```
}

if mouse_check_button_pressed(mb_left) &&
state="power"

{

  state="aim";


inst=instance_create_layer(x,y,"bullet",obj_bulle
t);

  inst.speed=pow/8;

  inst.image_angle=ang;

  inst.direction=ang;

  exit;

}
```

Draw Event:

```
draw_text(x,y+250,"Angle "+string(ang));

draw_text(x,y+300,"Power "+string(pow));

draw_self();
```

101 Creating Effect On Collision

Uses built in effects to create visuals when colliding.

Step Event:

```
x=mouse_x;
y=mouse_y;
inst=instance_place(x,y,obj_enemy);
if inst!=noone
{
  avx=x-((x-inst.x)/2);
  avy=y-((y-inst.y)/2);

effect_create_above(ef_spark,avx,avy,5,c_yellow);

effect_create_above(ef_flare,avx,avy,5,c_green);
  effect_create_above(ef_ring,avx,avy,5,c_white);
}
```